Reasoning about Time in the Situation Calculus.

Javier Pinto*
Depto. de Ciencia de la Computación
Pontificia Universidad Católica de Chile
Casilla 306, Santiago 22, Chile
jpinto@ing.puc.cl

Raymond Reiter†
Dept. of Computer Science
University of Toronto
Toronto, Canada M5S 1A4
reiter@ai.toronto.edu

Abstract

We extend the ontology of the situation calculus to provide for the representation of time and event occurrences. We do this by defining a time line corresponding to a sequence of situations (called *actual situations*) beginning with the initial situation. Actual situations are totally ordered and the actions that lead to different actual situations are said to have *occurred.* This exension to the situation calculus permits one to express truths about the state of the world at different times. For example, we can state that at some point in the future certain fluent will be true. We can also express constraints on the occurrences of events, for example, that after releasing a cup, it will eventually hit the floor.

Our version of the situation calculus subsumes other temporal logics. In particular, we show that the modal *Temporal Logic of Concurrency* [4] can be embedded in the extended situation calculus. Our extension can also realize the essential features of other first order proposals for reasoning about time commonly used for AI purposes (e.g., Allen [1], Kowalski and Sergot [6]).

1 Introduction and Motivation.

Although the situation calculus [11] is by now a standard formalism in AI for representing and reasoning about actions and their effects, it does not yet provide a very rich temporal ontology. There have been a few earlier proposals to add some temporal features to the situation calculus (e.g., [3, 20]), but these have not gone as far as one would like. In particular, as we describe below, it is impossible, in current versions of the situation calculus, to represent event or action *occurrences.*[1] This in turn prevents one from representing causality between events, for example, that an ignition-key turning event is always followed by an engine-starting event. Similarly, when reasoning about actions and time we often want to represent truths about fluents at a given time,

*This work was done while this author was with the Dept. of Computer Science, University of Toronto.

†Fellow of the Canadian Institute for Advanced Research

[1] In this paper, we do not distinguish between actions and events.

for example, that tomorrow morning it will be sunny, or that the car we parked outside last night is no longer there the following noon. There is no obvious way to represent such temporal information in the "classical" situation calculus.

On the other hand, knowledge of this kind can be expressed very easily in some linear temporal logics (e.g., [1, 21]). For example, in such logics, one can write sentences of the form $holdsAt(F, T)$ to express that fluent F is true at time T, and $occurs(A, T)$, meaning that the action A has occurred (or will occur) at time T. In this paper, we argue that this asymmetry between the situation calculus and linear time line logics is more apparent than real and we show how to enrich the situation calculus with these features.

To realize these extensions, we endow the branching structure of time, implicit in the situation calculus formalism, with a time line. For this purpose, we identify the *temporal line* of a linear temporal logic with a single path in the tree of situations. This single path will completely describe the world in a way that parallels the time line in a linear temporal logic. We present a formalization of the situation calculus enriched with a predicate *actual* that identifies a path of situations starting in S_0 (the initial situation). The path so identified describes the world's true evolution.

Modal temporal logics (e.g., [4]) are often encountered as alternatives to first order linear time logics like those in [1, 6]. In [11], McCarthy and Hayes conjecture that the expressive power of modal logics can be gained without the use of modalities using an ordinary truth-functional logic. Their proposal was to identify each "possible world" with a situation in a situation calculus language. In this paper, we pursue this suggestion, and show how the *modal temporal logic of concurrency* [4] can be embedded in the situation calculus. We do this by identifying the possible worlds with a fixed infinite branch of states in the situation calculus tree of states. This embedding shows that this modal logic is strictly less expressive than the extended situation calculus we propose. Of particular interest is the fact that none of the modal logics of which we are aware give first class status to events and their occurrences, which is a major asset of our situation calculus.

Independently, Miller and Shanahan ([12]) discuss some of the issues that are addressed in this paper. In particular, they discuss how to deal with *narratives* in the situation calculus. Miller and Shanahan define a narrative as an actual course of actions (or events) about which we may have incomplete information. They also deal with the problem of incorporating concurrent actions in the narratives, an issue we have not addressed in this paper.

2 The Language and the Axiomatization.

2.1 The Discrete Situation Calculus

We have sorts $\mathcal{A}, \mathcal{S}, \mathcal{F}, \mathcal{D}$ for action types, situations, fluents[2], and other domain objects respectively. Variables are denoted by lower case letters (with or without subscripts), and constants are denoted by upper case letters (with or without subscripts). Unless otherwise stated, letters a, s, f (A, S, F) are used for variables (constants) of sorts $\mathcal{A}, \mathcal{S}, \mathcal{F}$ respectively. We will have a single 1-place predicate variable φ over situations. Other variables and constants will be assumed to be of sort $\mathcal{D}$. Also, when free variables appear in formulas, they are assumed to be universally quantified.

Strictly speaking, our language is second order. However, the only second order sentence we use is the induction axiom (1). Therefore, we restrict ourselves to a standard sorted first order sublanguage (with equality) to express all other axioms of our theories.

[2] Fluents are those properties of the world that may change as a result of doing actions.

We include the special constant S_0 of sort $\mathcal{S}$, the function $do : \mathcal{A}\times\mathcal{S} \rightarrow \mathcal{S}$, the relation $< \subseteq \mathcal{S}\times\mathcal{S}$, along with other functions and relations introduced later.[3]

Basic Axioms:

$$(\forall\, \varphi).[\varphi(S_0) \wedge (\forall\, s, a)\ (\varphi(s) \supset \varphi(do(a, s)))] \supset (\forall\, s)\ \varphi(s), \tag{1}$$

$$(\forall\, s_1, s_2, a).s_1 < do(a, s_2) \equiv s_1 \leq s_2, \tag{2}$$

$$(\forall\, a_1, a_2, s_1, s_2).do(a_1, s_1) = do(a_2, s_2) \supset a_1 = a_2, \tag{3}$$

$$(\forall\, s_1, s_2).s_1 < s_2 \supset \neg\, s_2 < s_1. \tag{4}$$

Axiom (1) is an induction axiom, necessary to prove properties true in all situations. See Reiter [19] for examples of the need for, and use of, induction in the situation calculus.

Proposition 2.1 *Some consequences of these axioms are*[4]*:*

$$(\forall\, s)\ S_0 \leq s, \tag{5}$$

$$(\forall\, s)\ \neg s < S_0, \tag{6}$$

$$(\forall\, a, s)\ \neg\, s < s, \tag{7}$$

$$(\forall\, s).s \neq S_0 \supset (\exists\, a, s')\ s = do(a, s'), \tag{8}$$

$$(\forall\, s_1, s_2, s_3).s_1 < s_2 \wedge s_2 < s_3 \supset s_1 < s_3, \tag{9}$$

$$(\forall\, a, s)\ s < do(a, s), \tag{10}$$

$$(\forall\, a_1, a_2, s_1, s_2).do(a_1, s_1) < do(a_2, s_2) \supset s_1 < s_2, \tag{11}$$

$$(\forall\, a, s, s').do(a, s) \leq s' \supset s < s', \tag{12}$$

$$(\forall\, a, s_1, s_2).do(a, s_1) = do(a, s_2) \supset s_1 = s_2. \tag{13}$$

Now, we need to incorporate basic axioms introducing elements necessary to reason about the effects of actions, persistence, etc. To introduce change, *effect axioms* have to be specified; these have the following form:

$$Poss(A, s) \wedge \phi_{A,F}(s) \supset holds(F, do(A, s)). \tag{14}$$

Here, $\phi_{A,F}(s)$ is a first order formula providing qualifications for the effect of the action A on fluent F. $Poss(a, s)$ is a predicate that is true when action a can be performed in situation s.

In order to achieve the desired effects, we need to introduce some mechanism to enforce the notion of minimal change. Since the frame and ramification problems are not the primary focus of this paper, we will assume that either we have a complete monotonic specification of the theory of action (e.g., using Reiter's *successor state axioms* [18] to complete the set of effect axioms (14)); or some non-monotonic mechanism to *complete* the theory (e.g., Lin and Shoham [8]).

2.2 An Actual Path of Situations.

So far, we have introduced axioms that allow for the specification of what is true and what truths change along different paths that start in the initial situation S_0. Each one of these paths is

[3] We use the abbreviation $s_1 \leq s_2$ to mean $s_1 < s_2 \vee s_1 = s_2$. Also, we use the notation $a < b < c$ to mean $a < b \wedge b < c$, and similarly when the expression combines $<$ and $\leq$.

[4] All proofs are in the appendix.

interpreted as a different way in which the world could evolve. A limitation of the situation calculus view, which we address here, is that we have no way of describing how the world actually evolves. To this end, we incorporate a predicate *actual* for situations. The intended meaning is that a situation is *actual* if it lies on the path that describes the world's real evolution. The axioms for *actual* are:

$$actual(S_0), \tag{15}$$

$$(\forall\, a, s).actual(do(a, s)) \supset actual(s) \wedge Poss(a, s), \tag{16}$$

$$(\forall\, a_1, a_2, s).actual(do(a_1, s)) \wedge actual(do(a_2, s)) \supset a_1 = a_2. \tag{17}$$

Axioms (15-16) express that the initial situation is always actual, and if a situation is actual, then its immediate predecessor must also be. Axiom (17) says that an actual situation has at most one actual successor situation. An important characteristic of actual situations is that they all lie on the same path (i.e., they constitute a time line), as the following shows:

Proposition 2.2

$$(\forall\, s_1, s_2).s_2 \leq s_1 \supset [actual(s_1) \supset actual(s_2)],$$
$$(\forall\, s_1, s_2).actual(s_1) \wedge actual(s_2) \supset s_1 < s_2 \vee s_2 < s_1 \vee s_1 = s_2.$$

We also introduce a notion of time, which will allow us to establish a direct relationship between the situation calculus and linear temporal logics (e.g., the calculus of events [6]). Intuitively, each situation has a starting time and an ending time. During the time span of a situation no fluents change truth values[5]. We incorporate the sort $\mathcal{T}$, interpreted as a continuous time line, into our language and introduce the functions *start* from $\mathcal{S}$ to $\mathcal{T}$, and *end*, from $\mathcal{S} \times \mathcal{A}$ to $\mathcal{T}$. Events/actions occur at the ending time of situations. This is captured by the following axioms[6]:

$$(\forall\, s, a)\ end(s, a) = start(do(a, s)), \tag{18}$$

$$(\forall\, a, s)\ start(s) < end(s, a), \tag{19}$$

$$start(S_0) = 0. \tag{20}$$

These have the immediate consequence:

Proposition 2.3

$$(\forall\, s, s').s < s' \supset start(s) < start(s').$$

Occurrences are introduced as a relation between *event types* and situations. For example:

$$occurs(pickup(x), s)$$

says that a pickup event occurred in situation s. Occurrences are defined in terms of the actual path as follows:

$$occurs(a, s) \equiv actual(do(a, s)).$$

[5] This restricts the properties of the world that can be represented as fluents, for instance, the position of a moving ball cannot be represented by a fluent.

[6] In what follows, we use $<$ and $\leq$ as relations between situations (as defined earlier), as well as for the standard ordering relation between the elements of $\mathcal{T}$.

In some cases, it might be convenient to establish a relationship between events/actions that occur and the time at which they occur (rather than the situation). For this purpose, we introduce a predicate $occurs_{\mathcal{T}} \subseteq \mathcal{A} \times \mathcal{T}$, defined as:

$$occurs_{\mathcal{T}}(a, t) \equiv (\exists\, s).occurs(a, s) \wedge start(do(a, s)) = t.$$

Also, define a relation $holds_{\mathcal{T}}$ between fluents and time points and a relation *during* between time points and situations:

$$\begin{aligned}
&holds_{\mathcal{T}}(f, t) \equiv (\exists\, s).actual(s) \wedge during(t, s) \wedge holds(f, s),\\
&during(t, s) \equiv actual(s) \wedge start(s) < t \wedge (\forall\, a).[occurs(a, s) \supset end(s, a) \geq t].
\end{aligned}$$

Intra-state persistence is derivable:

Proposition 2.4

$$(\forall\, f, s, t, t').during(t, s) \wedge during(t', s) \supset holds_{\mathcal{T}}(f, t) \equiv holds_{\mathcal{T}}(f, t').$$

We can also introduce the notion of an event occurring between situations as follows:

$$occursBet(e, s_1, s_2) \equiv (\exists\, s).s_1 < s < s_2 \wedge occurs(e, s).$$

Proposition 2.5

$$actual(s_2) \wedge s_1 < s_2 \wedge \neg(\exists\, e)\ occursBet(e, s_1, s_2) \supset (\exists\, a)\ s_2 = do(a, s_1).$$

Notice that our treatment of $during(t, s)$ requires that s be *actual*; we have no way of expressing time point occurrences *during* non-actual situations. This would be necessary if we were to have "hypothetical actual" lines of situations. Unfortunately, we do not have an adequate account of hypothetical reasoning in the situation calculus, so we are postponing defining a fully general $during(t, s)$ until we do.

The expanded ontology for the situation calculus provides the essential features of the *event calculus* of Kowalski and Sergot [6], which in turn appears to be a variant of Allen's temporal logic [1]. See [16] for details.

2.3 On Actions with Durations.

In the previous section we formalized a notion of a time line within the situation calculus by identifying it with a connected path in the tree of situations. The structure of time is considered to be isomorphic to that of the positive real numbers. Each occurring action is associated with a time point at which the action is said to occur. Thus, elementary actions are seen as having instantaneous durations.

This point-like view of actions can be easily used to model actions that have durations. This is done in exactly the same manner as temporal intervals are modelled in point-based temporal ontologies. Thus, if an action has duration, we model it with two instantaneous actions *Start-action* and *End-action*. For example, consider the action of holding open a spring loaded lock of a door. This can be considered as an action with a temporal extent and would be modelled by introducing two action constants: *Start-holding* and *End-holding*. Furthermore, we introduce the fluent *Held-open* to indicate that the *holding* action is in progress. The relationship between this fluent and the instantaneous *start-end* actions is expressed using the effect axioms:

$$Poss(Start\text{-}holding, s) \supset holds(Held\text{-}open, do(Start\text{-}holding, s)),$$
$$Poss(End\text{-}holding, s) \supset \neg holds(Held\text{-}open, do(End\text{-}holding, s)).$$

Thus, if we assume that these sentences express all the reasons why *Held-open* may change, *Held-open* would be true exactly in the situations in which the lock is being held. Clearly, it is possible for other actions to occur after *Start-action* and before *End-action*. For example, the door can be pulled open in a situation in which the fluent *Held-open* is true. Therefore, we can model a restricted form of concurrency in which actions with durations occur concurrently as long as their starting and ending points don't coincide. This latter restriction can be eliminated by allowing for concurrent actions in the language of the situation calculus (see Gelfond et al. [3], Miller and Shanahan [12], Lin and Shoham [9] or Pinto [15] for examples on how to deal with concurrency in the situation calculus).

3 Some Applications.

Explaining Observations.

Assume that our car is parked in the morning:

$$holds(Parked, S_m) \wedge actual(S_m).$$

However, we observe some time later that the car is no longer parked:

$$actual(S_e) \wedge \neg holds(Parked, S_e) \wedge S_m < S_e.$$

Furthermore, assume that our theory includes the following successor state axiom (Reiter [18]):

$$\begin{aligned}Poss(a, s) \supset {} & holds(Parked, do(a, s)) \equiv \\ & [holds(Parked, s) \wedge a \neq Steal \wedge a \neq TowAway] \vee a = Park.\end{aligned}$$

Then, the following is a consequence of our axioms:

$$(\exists\, s).(S_m \leq s < S_e) \wedge [occurs(Steal, s) \vee occurs(TowAway, s)].$$

The issue of explaining observations in a similar framework is also addressed by Shanahan in [22].

Behavioural Constraints.

It is possible to express behavioural constraints for agents. For example: "Never cross the street against a red light:"

$$holds(Red, s) \supset \neg occurs(Cross, s).$$

"If you drink, don't drive:"

$$\neg occurs(Drive, do(Drink, s)).$$

We also have a natural notion of prevention. To prevent F from ever becoming false:

$$actual(s) \supset holds(F, s).$$

External Events.

We can formalize event occurrences not under an agent's control, but which can be used in planning. For example, assume that we need to describe the scheduling of a bus, which we know arrives every 15 minutes between 0800 and 2400. We also assume that the bus does not come at other times. Thus, we write:

$$occurs_{\mathcal{T}}(BusArrives, t) \equiv (\exists\, n).0 \leq n \leq 64 \wedge t = 800 + 100 \times n/4.$$

We can also say that the bus always waits for a minute before leaving:

$$occurs_{\mathcal{T}}(BusArrives, t) \supset occurs_{\mathcal{T}}(BusLeaves, t+1).$$

Independently of the description of external events, we can also describe the behaviour of an agent, possibly as the result of planning, using occurrences. For example:

$$occurs(AgentArrives, s) \wedge holds(BusPresent, s) \supset occurs(BoardsBus, do(AgentArrives, s)).$$

where *AgentArrives* and *BoardsBus* are actions performed by the agent. From a specification of the conditions under which the fluent *BusPresent* changes, we can derive its successor state axiom. Thus, we can we can easily derive that the agent boards the bus if he arrives at the bus stop before the last bus leaves.

Hypothetical Reasoning.

By preserving the branching state property of the situation calculus, we can express and answer a variety of hypothetical queries, although counterfactuals cannot be expressed. For example "At time T_p in the past, when you put A on B, could A have been put on C instead?" can be simply expressed as:

$$during(T_p, s) \wedge actual(s) \supset Poss(put(A, C), s).$$

"If in state S_p I had performed $put(A, C)$, would F have been true?"

$$holds(F, do(put(A, C), S_p)) \wedge Poss(put(A, C), S_p).$$

None of these features is possible in linear temporal logics. We need the branching structure of the situation calculus, coupled with a linear time line in that branching structure.

A shortcoming of the view of the actual line of situations discussed here is that we do not consider the existence of "hypothetical actual" branches. The present formalism allows us to express knowledge about what happens in the world, or what Miller and Shanahan call "narratives." Unfortunately, we do not have hypothetical actual lines that would allow us to express knowledge about what would occur if certain non-actual actions were to be performed, for example, counterfactuals in which alternative action occurrences can be postulated. These are issues which we are currently investigating, but about which we do not yet have a definitive proposal.

Chains of events.

In many cases several events occur in sequences. The events in these sequences may be related to one another by *causal* relationships. For example, *shooting a loaded gun causes a noise to occur*:

$$holds(Loaded, s) \wedge occurs(Shoot, s) \supset occurs(Noise, do(Shoot, s)), \qquad (21)$$

or, *everything that goes up must come down*:

$$occurs(GoesUp, s) \supset occurs(ComesDown, do(GoesUp, s)). \qquad (22)$$

According to sentences (21) and (22), no events may occur between the first and the second actions mentioned. If this is not desired, simply leave the intervening action occurrences open:

$$occurs(GoesUp, s) \supset (\exists\, s').s < s' \wedge occurs(ComesDown, s').$$

4 The Situation Calculus and Modal Temporal Logics.

In [11], McCarthy and Hayes argue, without proof, that the expressive power of modal logics can be gained without the use of modalities using an ordinary truth-functional logic. The proposal is to identify each "possible world" with a situation in a situation calculus language. In this section we lend support to this intuition by showing how one particular such logic – the Temporal Logic of Concurrency – can be embedded in the extended situation calculus. The advantage of performing this embedding is that the expressiveness of the temporal logic is preserved, while avoiding the use of "mixed logics", in much the same way as does Moore [13] in providing an embedding of logics of knowledge into the situation calculus. Moreover, we retain the property that actions are first class citizens in the situation calculus, something that modal temporal logics lack.

4.1 The Temporal Logic.

The *Temporal Logic of Concurrency* (*TL* for short) [4] is a propositional modal logic, in which the "possible worlds" are temporally ordered *states*. *TL* formulas are defined in terms of atomic propositions by the following BNF expression:

$$A ::= f \mid \neg A \mid A \rightarrow A \mid \Box A \mid \bigcirc A \mid A\, \mathcal{U}\, A$$

Informally, the meanings of the sub-expressions are as follows: f represents an elementary proposition that is true *now*; $\Box A$ means A is true from now on. $\bigcirc A$ means, true at the next state. $A\, \mathcal{U}\, B$ means A is true until B is true; from then on, A is unrestricted. The formula $\neg\Box\neg A$ is abbreviated as $\Diamond A$[7]. A model $\mathcal{M}$ of a *TL* theory is a structure $\langle S, \sigma, V\rangle$, in which S is an enumerable set of states, σ is a surjective function from the non-negative integers to S (σ_i denotes the state resulting from applying σ to the non-negative integer i). The sequence $\sigma_0 \ldots \sigma_i \ldots$ represents a temporally ordered sequence of states. V is a function from the atomic formulas or propositions to a subset of states in which the atomic formulas are *true*. $\mathcal{M} \models_i A$ means that A is true in the state σ_i; $\mathcal{M} \models A$ means that A is true in every state. For a complete definition of the $\models_i$ relation see [4, p.72].

Axiom Schemata of *TL*:

$$\Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B),$$
$$\bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B),$$

[7] The other usual logical connectives ($\vee$, $\wedge$, $\leftrightarrow$) are defined in terms of negation ($\neg$) and implication ($\rightarrow$) in the standard way.

$$\begin{aligned}
&\bigcirc\neg A \leftrightarrow \neg \bigcirc A,\\
&\Box A \rightarrow A \wedge \bigcirc\Box A,\\
&\Box(A \rightarrow \bigcirc A) \rightarrow (A \rightarrow \Box A),\\
&A\,\mathcal{U}\,B \rightarrow \Diamond B,\\
&A\,\mathcal{U}\,B \leftrightarrow B \vee (A \wedge \bigcirc(A\,\mathcal{U}\,B)).
\end{aligned} \tag{23}$$

TL also includes all tautologies.

Notice that (23) is an induction axiom.

Rules of Inference of TL:

- Detachment: from A and $A \rightarrow B$ infer B.
- Necessitation: from A infer $\Box A$ and $\bigcirc A$.

4.2 The Embedding.

It is important to observe that even though the logic just presented is called a *temporal logic of concurrency*, it does not provide a mechanism for dealing with true *concurrency*. In fact, in this setting, concurrency means that there might be multiple agents co-existing but it is not possible for them to execute actions in a concurrent manner (i.e., simultaneously). Therefore, we do not need true concurrent actions to embed *TL* in the situation calculus.

In order to perform the embedding, we give a translation of *TL* formulas to the situation calculus, and also provide a correspondence between the notions of $\models_i$ and $\models$ in *TL* and entailment in our version of the situation calculus. Since *TL* takes time to be linear, we draw a correspondence between the states σ_i and the states of the *actual* line. Furthermore, in what follows, all our axioms refer to *actual* states.

Syntactic Characterization. Each atomic formula in *TL* is characterized by a constant of sort fluent ($\mathcal{F}$). Let Λ be an arbitrary formula in *TL*. Its translation into the situation calculus is:

$$(\forall\, s).actual(s) \supset holds(\Lambda, s).$$

where $holds(\Lambda, s)$ abbreviates a situation calculus formula defined as follows[8]:

$$\begin{aligned}
&holds(\neg\Lambda, s) = \neg holds(\Lambda, s)\\
&holds(\Lambda_1 \rightarrow \Lambda_2, s) = holds(\Lambda_1, s) \supset holds(\Lambda_2, s)\\
&holds(\Box\Lambda, s) = (\forall\, s').s' \geq s \wedge actual(s') \supset holds(\Lambda, s')\\
&holds(\bigcirc\Lambda, s) = (\forall\, a).actual(do(a, s)) \supset holds(\Lambda, do(a, s))\\
&holds(\Lambda_1\,\mathcal{U}\,\Lambda_2, s) = (\exists\, s').s' \geq s \wedge actual(s') \wedge holds(\Lambda_2, s') \wedge\\
&\qquad\qquad (\forall\, s'')\ [s \leq s'' < s' \supset holds(\Lambda_1, s'')]
\end{aligned}$$

It is trivial to show that the abbreviation for $holds(\neg\Box\neg\Lambda, s)$ is equivalent to the intuitively correct expression:

$$(\exists\, s').s' \geq s \wedge actual(s') \wedge holds(\Lambda, s').$$

[8] The translation of formulas involving the defined logical connectives ($\vee$, $\wedge$, $\leftrightarrow$) is straightforward.

Semantic Concerns. Models in *TL* are structures in which the function σ defines an infinite sequence of states. The intuition behind this sequence is that the world passes through each state in this sequence in the order prescribed by it. In order to ensure that our models contain an infinite sequence of actual situations, we need to add the following axiom:

$$(\forall\, s).actual(s) \supset (\exists\, a).actual(do(a,s)). \tag{24}$$

This axiom says that every actual situation must have a *next* actual situation, which, given axiom (17), is unique.

We must show that the translation of each instance of an axiom schema of *TL* is entailed by our enriched situation calculus.

Proposition 4.1 *If* Λ *is an instance of an axiom schema of* TL*, then*

$$\Sigma \models (\forall\, s).actual(s) \supset holds(\Lambda, s),$$

where Σ *is the set of situation calculus axioms (1),(2),(4), (15), (17) and (24).*

For most *TL* schemata, the proof is straightforward, but somewhat tedious. Perhaps, the most interesting proof is that of the *TL* induction schema (23) which, not surprisingly, requires the use of our situation calculus induction axiom (1).

Also:

Proposition 4.2 *The necessitation rules for* $\Box$ *and* $\bigcirc$ *are sound, since it is trivial to show that if* Σ *includes the situation calculus axioms (1),(2),(4), (15), (17) and (24):*

$$\Sigma \models (\forall\, s).actual(s) \supset holds(\Lambda, s) \;\Rightarrow\; \Sigma \models (\forall\, s).actual(s) \supset holds(\Box\Lambda, s),$$

as well as:

$$\Sigma \models (\forall\, s).actual(s) \supset holds(\Lambda, s) \;\Rightarrow\; \Sigma \models (\forall\, s).actual(s) \supset holds(\bigcirc\Lambda, s).$$

Furthermore, let Φ_{TL} be an arbitrary set of *TL* sentences, and let ϕ be a sentence of *TL*. If Φ_{sc} corresponds to the translation of Φ_{TL} to the situation calculus, then it is not difficult to see that $\Phi_{TL} \models \phi$[9] iff $\Sigma \cup \Phi_{sc} \models (\forall\, s).actual(s) \supset holds(\phi, s)$.

Finally, it is also the case that $\Phi_{TL} \models_i \phi$ iff:

$$\Sigma \cup \Phi_{sc} \models (\forall\, a_1,\ldots,a_i).actual(do([a_1,\ldots,a_i],S_0)) \supset holds(\phi, do([a_1,\ldots,a_i],S_0)), \tag{25}$$

where $do([a_1,\ldots,a_i],S_0)$, denotes the situation term:

$$do(a_i, do(a_{i-1},\ldots do(a_1,S_0)\ldots)).$$

Since, by axiom (17), for any i there exists a unique sequence $[a_1,\ldots,a_i]$ of actions such that $do([a_1,\ldots,a_i],S_0)$ is actual, we can write (25) as:

$$\Sigma \cup \Phi_{sc} \models (\exists\, a_1,\ldots,a_i).actual(do([a_1,\ldots,a_i],S_0)) \wedge holds(\phi, do([a_1,\ldots,a_i],S_0)),$$

We conclude that every entailment of a *TL* theory is an entailment of its situation calculus translation.

In summary:

Theorem 4.1 *The extended situation calculus subsumes the temporal logic of concurrency.*

[9] $\Phi_{TL} \models \phi$ means that ϕ is true in all the states in all models of Φ_{TL}

4.3 Further Comments.

We have embedded the modal logic *TL* in the situation calculus. The question arises: Is it possible to realize this embedding without appealing to the extended situation calculus? The answer is yes. However, the resulting language is greatly impoverished. To see why, consider the axiom schema:

$$\bigcirc \neg A \leftrightarrow \neg \bigcirc A,$$

which ensures that $\bigcirc$ acts as a functional operator, i.e., every state has exactly one successor. We achieved this by mapping *TL* states to the extended situation calculus' *actual situations*, which also have a unique successor situation in the actual line. This can also be realized in the standard situation calculus by allowing only one action. In this way, every situation would have exactly one successor. The problem with this solution is that it gives up the ability to talk about arbitrary actions. If only one action exists (whose only purpose is to produce a state change), then we lose the ability to represent different actions and their effects.

With our approach, on the other hand, not only can we gain the same expressiveness of the *TL* modal logic, but we retain the basic features of the original situation calculus, plus the ability to talk about event occurrences. The only limitation introduced is that axiom (24) prevents us from expressing *deadlock* situations, i.e., situations in which no actions are possible. This is not a serious limitation since we can achieve the effect of a deadlock by introducing a *Wait* action with no effects.

5 Conclusions and Future Research.

5.1 What Has Been Achieved?

- The standard alternatives to the situation calculus in the AI literature (e.g., Allen [1], Kowalski and Sergot [6]) are linear time theories which provide for the representation of event occurrences and fluent truths at time t. We have shown how, with a minimal change to its ontology, the situation calculus can realize the essential features of these alternatives, without sacrificing its capacity for deductive planning and hypothetical reasoning provided by its branching state structure.

- As we discuss in [16], there is a close correspondence between our enriched situation calculus and the *calculus of events* of Kowalski and Sergot. One advantage of this is the clean semantics provided by our axiomatization, in contrast to the event calculus's reliance on the *negation as failure* feature of logic programming, whose semantics is not well understood. Moreover, as we show in [16], our axioms lead to a logic programming implementation with comparable computational properties to those of the event calculus.

- We have confirmed the intuitions of McCarthy and Hayes [11] that the situation calculus is at least as expressive as modal temporal logics. This we did by providing a translation of the logic *TL* into the extended situation calculus. This avoids the problem of "mixed logics", in much the same way as does Moore [13] in providing an embedding of logics of knowledge into the situation calculus. Moreover, by embedding the essential features of modal temporal logic within the situation calculus, we preserve the action-based ontology of the situation calculus; conventional temporal logics do *not* provide for actions and their effects in their ontologies.

- All application domains for which temporal logics have been used can be studied within the framework of the extended situation calculus (e.g., semantics of programming languages [7], reasoning about digital circuits [14, 10], modelling of concurrency [17], etc.). This provides the potential for integrating results in hardware specification (e.g., [10]), with research in fault diagnosis [5]. Among other things, this could lead to a rich theory of diagnosis for dynamic systems.

5.2 Future Research.

- **Enriched event ontologies.** The axiomatization of this paper assumes non-concurrent, discrete events. These limitations need to be removed, perhaps by appealing to ideas of Lin and Shoham [9] and Gelfond, Lifschitz and Rabinov [3].

- **Foundations for reasoning about physical processes.** Provided we can represent continuity within the extended situation calculus, we will be able to describe processes like: "If I provide a continuous supply of heat to a pot of water, the water will eventually boil." A language of this sort is necessary for formally specifying systems that reason about physical processes, as in (Davis [2]).

- **Planning with external events.** Conventional planning techniques assume that all actions are under the agent's control. The possibility of an agent exploiting external events in its plans needs to be addressed.

- **Time and complex events.** The approach of this paper supposes that all events are primitive. Levesque, Lin and Reiter[10] have augmented the "classical" situation calculus with the ability to define complex events from primitive ones using quantification, sequencing, conditionals, nondeterministic choice and iteration. Augmenting this theory with a time line would provide all the benefits we have described in this paper, but generalized to this much richer language.

- **Relationship with other frameworks for temporal reasoning.** It would be interesting to investigate how other temporal logics can be embedded in the situation calculus, and how their built-in assumptions about the structure of time translate to the situation calculus view of time.

- **Sophisticated hypothetical reasoning.** While the approach of this paper supports some forms of hypothetical reasoning (Section 3), it is not completely general. One cannot, for example, pose and answer counterfactual queries like: "If I had not paid the bookie, would I have lived to bet again." To handle such examples, it appears to be necessary to hypothesize a different time line than the actual one, but with the same properties as the actual time line. It is in reasoning of this kind where the branching state structure of the situation calculus becomes particularly important.

[10] Forthcoming.

A Proofs

Proof of Proposition 2.1.

1. $(\forall s). S_0 \leq s$. By induction, using $\varphi(s) = S_0 \leq s$.
2. $(\forall s)\ \neg s < S_0$. Use (4) and (5).
3. $(\forall a, s). \neg\, s < s$. By contradiction, using (4) with $s_1 = s_2 = s$.
4. $(\forall s). s \neq S_0 \supset (\exists\, a, s'). s = do(a, s')$. By induction, with $\varphi(s) = [s \neq S_0 \supset (\exists\, a, s'). s = do(a, s')]$.
5. $(\forall s_1, s_2, s_3). s_1 < s_2 \wedge s_2 < s_3 \supset s_1 < s_3$. By induction, with $\varphi(s) = (\forall s_1, s_2). s_1 < s_2 \wedge s_2 < s \supset s_1 < s$.
6. $(\forall a, s). s < do(a, s)$. Use (2).
7. $do(a_1, s_1) < do(a_2, s_2) \supset s_1 < s_2$. Use (2), (9) and (10).
8. $(\forall a, s, s'). do(a, s) \leq s' \supset s < s'$. Use (9), (10) and (11).
9. $(\forall a, s_1, s_2). do(a, s_1) = do(a, s_2) \supset s_1 = s_2$. By contradiction, using (2), (4) and (10).

Proof of Proposition 2.2: Use the induction axiom (1). The first part of the proposition follows by taking:

$$\varphi(s) = (\forall s'). s' \leq s \supset [actual(s) \supset actual(s')],$$

and the second part by taking:

$$\varphi(s) = (\forall s'). actual(s) \wedge actual(s') \supset s < s' \vee s' < s \vee s = s'.$$

Proof of Proposition 2.3: Use the induction axiom (1) with:

$$\varphi(s) = (\forall s'). s' < s \supset start(s') < start(s).$$

Proof of Propositions 2.4 and 2.5: Both follow directly from the definitions of $holds_{\mathcal{T}}$, *during* and *occursBet*.

Proof of Proposition 4.1: We must prove that:

$$\Sigma \models (\forall s). actual(s) \supset holds(\Lambda, s), \tag{26}$$

for each instance Λ of an axiom schema of TL.

1. $\Lambda = \Box(A \rightarrow B) \rightarrow (\Box A \rightarrow \Box B)$. The translation of Λ is a valid first order sentence.
2. $\Lambda = \bigcirc(A \rightarrow B) \rightarrow (\bigcirc A \rightarrow \bigcirc B)$. The translation of Λ is a valid first order sentence.
3. $\Lambda = \bigcirc\neg A \leftrightarrow \neg \bigcirc A$. Use axioms (17) and (24).
4. $\Lambda = \Box A \rightarrow A \wedge \bigcirc\Box A$. Use (12).

5. $\Lambda = \Box(A \rightarrow \bigcirc A) \rightarrow (A \rightarrow \Box A)$. After invoking some of the translation rules, Λ can be seen to be logically equivalent to:

   $(\forall s').(\forall s).actual(s) \wedge holds(\Box(A \rightarrow \bigcirc A), s) \wedge holds(A, s) \wedge s' \geq s \wedge actual(s') \supset holds(A, s')$.

   Now use the situation calculus induction axiom (1) with

   $\varphi(s') = (\forall s).actual(s) \wedge holds(\Box(A \rightarrow \bigcirc A), s) \wedge holds(A, s) \wedge s' \geq s \wedge actual(s') \supset holds(A, s')$.

   (a) The base case follows using axiom (15) and (6).

   (b) The inductive case follows using axiom (2), (10) and the first sentence of Proposition 2.2.

6. $\Lambda = A\ \mathcal{U}\ B \rightarrow \Diamond B$. The translation of Λ is a valid first order sentence.

7. $\Lambda = A\ \mathcal{U}\ B \leftrightarrow B \vee (A \wedge \bigcirc(A\ \mathcal{U}\ B))$. First, using (2), (7) and the second sentence of Proposition 2.2, prove the following:

   $$actual(s') \wedge actual(do(a, s)) \wedge s < s' \supset do(a, s) \leq s'. \qquad (27)$$

   (a) The proof of the required result from left to right follows from (27).

   (b) The proof from right to left follows from (27), (9), (12), the first sentence of Proposition 2.2 and axiom (24).

□

Acknowledgements

We are grateful to Leopoldo Bertossi, Michael Gruninger, Todd Kelley, Hector Levesque, Fangzhen Lin, Sheila McIlraith, Dale Schuurmans and Steven Shapiro for helpful discussions on the topic of this paper. We also thank the attendants of *The Second Symposium on Logical Formalizations of Commonsense Reasoning* for their valuable feedback on some aspects of this work. This research was funded by the National Science and Engineering Research Council of Canada, and the Institute for Robotics and Intelligent Systems.

Afterword (by Ray Reiter)

It is a great pleasure for me to contribute to this volume honouring Jack Minker's distinguished career, especially because my very first publications in AI and databases appeared in the 1978 pioneering book *Logic and Databases* which Jack co-edited with Hervé Gallaire. In fact, Jack was in many ways responsible for my participation in this area; it was Jack who encouraged my initial interests by inviting me to join the workshop that eventually lead to that book, and it was Jack who generously promoted my later efforts along these lines. I believe it's safe to say that no single person has contributed more than he has to the definition, nourishment and technical agenda of databases and their relations to logic, logic programming and artificial intelligence, and for this we all owe Jack a great debt. So also do the many scientists around the world whose human rights Jack has so vigorously defended through his efforts with the ACM and the American Association for the Advancement of Science. Thus on this occasion I am delighted to be able to salute Jack for his fundamental contributions to science and human decency. He is, to invoke an expression I've often heard him use, a real *mensch*, who does honour to us all.

References

[1] Allen, J. F. Towards a general theory of action and time. *Artificial Intelligence 23* (1984), 123–154.

[2] Davis, E. Axiomatizing qualitative process theory. In *Proceedings of the Third International Conference on Principles of Knowledge Representation and Reasoning (KR'92)* (1992), C. R. B. Nebel and W. Swartout, Eds., Morgan Kaufmann Publishers, Inc., pp. 177–188.

[3] Gelfond, M., Lifschitz, V., and Rabinov, A. What are the Limitations of the Situation Calculus? In *Working Notes, AAAI Spring Symposium Series. Symposium:Logical Formalization of Commonsense Reasoning.* (Mar. 1991), pp. 59–69.

[4] Goldblatt, R. *Logics of Time and Computation.* CSLI, 1987.

[5] J. de Kleer, Mackworth, A., and Reiter, R. Characterizing Diagnoses. In *Proceedings of the eighth National Conference on Artificial Intelligence (AAAI-90)* (1990), pp. 324–330.

[6] Kowalski, R., and Sergot, M. A logic-based calculus of events. *New Generation Computing 4* (1986), 67–95.

[7] Kröger, F. *Temporal Logic of Programs*, vol. 8 of *EATCS Monographs on Theoretical Computer Science.* Springer Verlag, 1987.

[8] Lin, F., and Shoham, Y. Provably Correct Theories of Action (preliminary report). In *Proceedings of the ninth National Conference on Artificial Intelligence (AAAI-91)* (1991), pp. 349–354.

[9] Lin, F., and Shoham, Y. Concurrent Actions in the Situation Calculus. In *Working Notes of the 4th International Workshop on Nonmonotonic Reasoning* (1992), pp. 133–138.

[10] Maruyama, F., and Fujita, M. Hardware Verification. *IEEE Computer* (February 1985), 22–32.

[11] McCarthy, J., and Hayes, P. J. Some philosophical problems from the standpoint of artificial intelligence. In *Machine Intelligence* 4, B. Meltzer and D. Michie, Eds. Edinburgh University Press, Edinburgh, Scotland, 1969, pp. 463–502.

[12] Miller, R., and Shanahan, M. Narratives in the situation calculus. Tech. rep., Imperial College, Department of Computing, 1993.

[13] Moore, R. C. A formal Theory of Knowledge and Action. In *Formal Theories of the Commonsense World*, J. B. Hobbs and R. C. Moore, Eds. Ablex Publishing Corp., Norwood, New Jersey, 1985, ch. 9, pp. 319–358.

[14] Moszkowski, B. *Reasoning About Digital Circuits.* PhD thesis, Department of Computer Science, Stanford University, 1983.

[15] Pinto, J. *Temporal Reasoning in the Situation Calculus.* PhD thesis, Department of Computer Science, University of Toronto, Toronto, Ontario, Canada, Feb. 1994.

[16] Pinto, J., and Reiter, R. Temporal reasoning in logic programming: A case for the situation calculus. In *Proceedings of the Tenth International Conference on Logic Programming* (Budapest, June 1993).

[17] Pnueli, A. The temporal semantics of concurrent programs. *Theoretical Computer Science 13* (1981), 45–60.

[18] Reiter, R. *The Frame Problem in the Situation Calculus: A Simple Solution (sometimes) and a completeness result for goal regression.* Artificial Intelligence and Mathematical Theory of Computation: Papers in Honor of John McCarthy. Academic Press, San Diego, CA, 1991, pp. 359–380.

[19] Reiter, R. Proving Properties of States in the Situation Calculus. *Artificial Intelligence 64*, 2 (1993), 337–351.

[20] Schubert, L. Monotonic solution of the frame problem in the situation calculus: an efficient method for worlds with fully specified actions. In *Knowledge Representation and Defeasible Reasoning*, H. Kyberg, R. Loui, and G. Carlson, Eds. Kluwer Academic Press, 1990, pp. 23–67.

[21] Shanahan, M. A circumscriptive calculus of events. Tech. rep., Imperial College, Department of Computing, 1992.

[22] Shanahan, M. Explanation in the situation calculus. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-93)* (1993). To appear.